Experiments and Analyses of Data Transfers Over Wide-Area Dedicated Connections

Nageswara S. V. Rao, Qiang Liu, Satyabrata Sen, and Jesse Hanley
Computational Sciences and Engineering Division
Oak Ridge National Laboratory
Oak Ridge, TN 37831
{raons,liuq1,sens,hanleyja}@ornl.gov

Ian Foster and Rajkumar Kettimuthu
Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, IL 60439
{foster,kettimut}@anl.gov

Chase Q. Wu and Daqing Yun
Department of Computer Science
New Jersey Institute of Technology
Newark, NJ 07102
{chase.wu,dy83}@njit.edu

Don Towsley and Gayane Vardoyan
Department of Computer Science
University of Massachusetts at Amherst
Amherst, MA 01003
{towsley,gvardoyan}@cs.umass.edu

***Abstract*—Dedicated wide-area network connections are increasingly employed in high-performance computing and big data scenarios. One might expect the performance and dynamics of data transfers over such connections to be easy to analyze due to the lack of competing traffic. However, non-linear transport dynamics and end-system complexities (e.g., multi-core hosts and distributed filesystems) can in fact make analysis surprisingly challenging. We present extensive measurements of memory-to-memory and disk-to-disk file transfers over 10 Gbps physical and emulated connections with 0–366 ms round trip times (RTTs). For memory-to-memory transfers, profiles of both TCP and UDT throughput as a function of RTT show concave and convex regions; large buffer sizes and more parallel flows lead to wider concave regions, which are highly desirable. TCP and UDT both also display complex throughput dynamics, as indicated by their Poincaré maps and Lyapunov exponents. For disk-to-disk transfers, we determine that high throughput can be achieved via a combination of parallel I/O threads, parallel network threads, and direct I/O mode. Our measurements also show that Lustre filesystems can be mounted over long-haul connections using LNet routers, although challenges remain in jointly optimizing file I/O and transport method parameters to achieve peak throughput.**

***Index Terms*—Wide-area transport, dedicated connections, TCP, UDT, throughput and file I/O profiles, throughput dynamics, Poincaré map, Lyapunov exponent.**

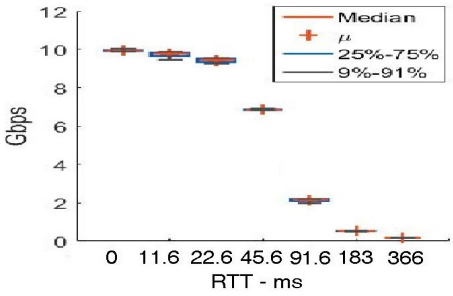
I. Introduction

There have been unprecedented increases in the volume and variety of datasets being processed by multi-site cloud computing server complexes, high-performance computing (HPC) workflows, and distributed big data computing facilities. The increasingly distributed nature of the underlying computations makes it important to be able to transfer large datasets effectively across long distances. Advanced data movement capabilities are often required, ranging from the high-throughput transfer of entire files to exchange of partial computation results, in all cases with stable and predictable dynamics. For example, effective memory-to-memory transfers are critical for applications such as coordinated computations over cloud servers at geographically dispersed sites, and for on-going computations on supercomputers steered by remote analysis and visualization codes. Traditional shared IP networks have not worked well for these specialized scenarios due to unpredictable loads from competing traffic flows. In response, dedicated connections are becoming increasingly deployed in science and commercial environments.

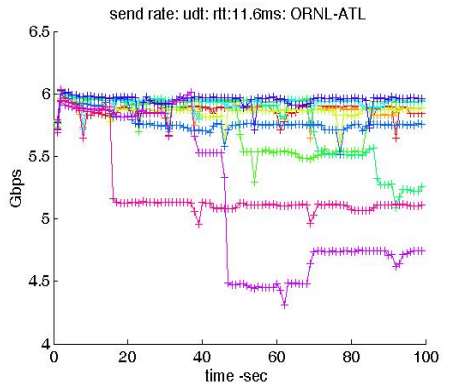
Specifically, high-performance network infrastructures such as the Department of Energy's (DOE) ESnet [6] and Google's Software Defined Network (SDN) [10] provide on-demand, dedicated connections. Within the DOE HPC infrastructure, special-purpose Data Transfer Nodes (DTNs) [5] are installed to take advantage of dedicated circuits [17] provisioned over ESnet. Furthermore, Lustre over Ethernet enables filesystems to be mounted across long-haul links, thereby overcoming the 2.5 ms latency limitation of Infiniband (IB) [26]; this approach provides wide-area file access without the use of special tools such as GridFTP [29] or XDD [21], [31], or hardware IB range extenders [1], [16]. Transmission Control Protocol (TCP) and to a lesser extent User Datagram Protocol (UDP) (with suitable loss recovery and fairness enhancements) are used for wide-area data transfers, including over dedicated connections.

It is generally expected that the performance dynamics of dedicated connections should be simpler than those of shared connections, and that the underlying data flows should achieve high throughput for large data transfers and stable dynamics for control and steering operations. However, experimental and analytical studies of such flows are quite limited, since most research has focused on shared network environments [28]. Our work has shown that while parameter optimizations specific to dedicated connections are indeed somewhat simpler than those required for shared connections, they are not simple extensions of the well-studied shared connection solutions [12], [21]. This observation led to the establishment of a testbed at Oak Ridge National Laboratory (ORNL) and the development of measurement and analysis methods for these scenarios.

We conduct structured experiments of both memory-to-memory and disk-to-disk transfers using the testbed. For



(a) Throughput profile $\Theta_O(\tau)$ for single STCP flow [22]



(b) Time trace $\theta(\tau, t)$ of UDT memory-to-memory transfer [12]

Fig. 1: Throughput profile of Scalable-TCP and time traces of UDT

memory-to-memory transfers, we consider the UDP-based Data Transfer Protocol (UDT) [7] and four TCP variants, namely, CUBIC [24], Hamilton TCP (HTCP) [27], BBR [4], and Scalable TCP (STCP) [11], which are among those considered suitable for high-bandwidth connections. For disk-to-disk transfers, we study XDD [31] and also mount the Lustre filesystem over long-haul connections. We use dedicated physical connections and a suite of hardware-emulated 10 Gbps connections with 0–366 ms round-trip times (RTTs).

For a given configuration of end-systems, hosts, transport method, and connection parameters, we denote the throughput at time t over a connection of RTT τ as $\theta(\tau, t)$. Its average over an observation period T_O is called the *throughput profile*:

$$\Theta_O(\tau) = \frac{1}{T_O} \int_0^{T_O} \theta(\tau, t)\, dt$$

Fig. 1(a) shows a representative plot of mean throughput profiles for a single STCP stream [22]. It is *concave* for lower RTTs and *convex* for higher RTTs. These dual-mode profiles are observed for both TCP [23] and UDT [12]. The concave regions are highly desirable since the throughput decreases slowly as RTT increases and in particular is higher than linear interpolation of measurements. Interestingly, such profiles are in stark contrast both to those predicted by analytical models and to experimental measurements collected for various TCP variants [8], [20], [32] and UDT [7]. The conventional TCP models lead to entirely convex profiles where the throughput decreases faster with RTT [14], [18], [28], and do not explain this dual-regime profile. Our measurements demonstrate that both large host buffers and an increasing number of parallel streams expand the concave regions, in addition to improving throughput.

Throughput time traces also exhibit rich dynamics, as illustrated in Fig. 1(b) for memory-to-memory transfers using UDT. These dynamics impact the throughput profiles in subtle ways, as revealed by our application of Poincaré map and Lyapunov exponent methods from chaos theory [22]. For TCP, at lower RTTs, higher throughput and smaller variations result in concave regions, and at higher RTTs, lower throughput and larger variations lead to convex regions.

This paper presents a comprehensive view of measurements and analyses of systematic experiments conducted using our testbed. It unifies the analysis of throughput profiles of TCP [22], UDT [12], and XDD [21], which we have previously treated separately but in more detail. We also present newer throughput profiles of BBR and compare them with those of HTCP, STCP, and CUBIC in [22]. Additionally, we include some preliminary throughput profiles of the Lustre filesystem mounted over long-haul Ethernet connections.

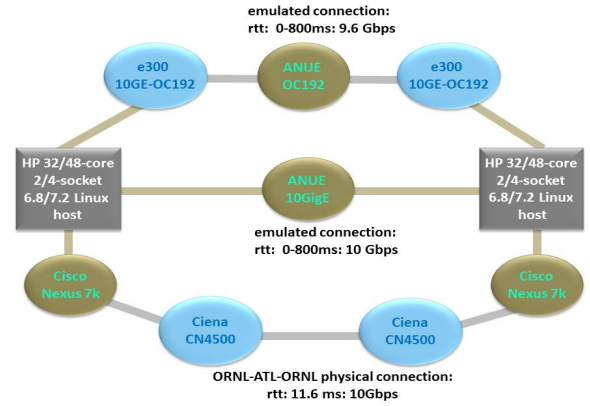
The organization of this paper is as follows. A brief description of our testbed is provided in Section II. The concave and convex regions of TCP and UDT throughput profiles are discussed in Section III. Analysis of transport dynamics using Poincaré map and Lyapunov exponents is briefly presented in Section IV. File transfer measurements are described in Section V. Conclusions and open areas are briefly described in Section VI.

II. Network Testbed with FileSystems

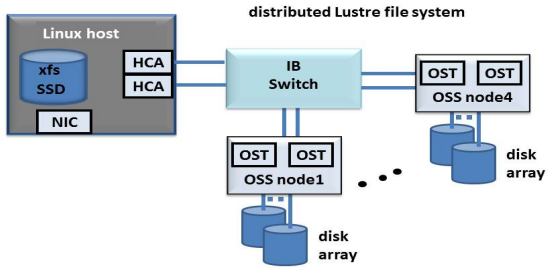
We performed experiments over a testbed consisting of 32/48-core Linux workstations, 100G Ethernet local and long-haul switches, and QDR IB switches, 10 Gbps hardware connection emulators, and disk controllers. The testbed is located at ORNL, and is also connected to a 500 mile 100 Gbps physical loop-back connection to Atlanta. For various network connections, hosts with identical configurations are connected in pairs over a back-to-back fiber connection with negligible 0.01 ms RTT and a physical 10GigE connection with 11.6 ms RTT via Cisco and Ciena devices, as shown in Fig. 2(a). The 10GigE and SONET/OC192 connections represent different physical modalities; we present results with the latter.

We use ANUE devices to emulate 10GigE and SONET/OC192 connections with RTTs $\tau \in \{0.4, 11.8, 22.6, 45.6, 91.6, 183, 366\}$ ms. We chose these RTT values to represent three basic scenarios: lower values represent cross-country connections, for example, between facilities across the US; 93.6 and 183 ms represent inter-continental connections, for example, between US, Europe, and Asia; and 366 ms represents a connection spanning the globe and is mainly used as a limiting case.

Memory-to-memory throughput measurements for TCP are collected using *iperf* and for UDT using its custom codes. Typically, 1-10 parallel streams are used for each configuration,

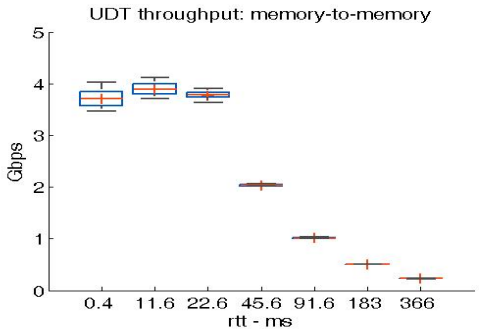


(a) Physical and emulated connections between hosts

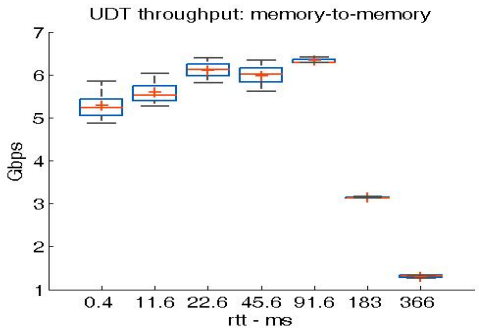


(b) Lustre and XFS filesystems

Fig. 2: Testbed network connections and filesystems



(a) default



(b) jumbogram

Fig. 3: Aggregate throughput profiles for UDT

and throughput measurements are repeated ten times. Three different TCP buffer sizes are used: default, normal (recommended values for 200 ms RTT), and large (the largest size allowed by the host kernel); and the socket buffer parameter for iperf is 2 GB. These settings result in the allocation of 250 KB, 250 MB, and 1 GB socket buffer sizes, respectively.

The Lustre filesystem employs multiple Object Storage Targets (OSTs) to manage collections of disks, multiple Object Storage Servers (OSSes) to stripe file contents, and distributed MetaData Servers (MDSes) to provide site-wide file naming and access. Our testbed consists of a distributed Lustre filesystem supported by eight OSTs connected over IB QDR switch as shown in Fig. 2(b). Host systems are connected to IB switch via Host Channel Adapters (HCA) and Luster over IB clients are used to mount the filesystem on them. In addition, our Solid-State Storage Device (SSD) drives are connected over PCI buses on the hosts, which mount local XFS filesystems. Data transfers between filesystems locally mounted at sites and connected over long-haul links are carried out using XDD [31] whose codes are used for throughput measurements. We also consider configurations wherein Lustre is mounted over long-haul connections using LNet routers, and in this case we utilize IOzone for throughput measurements for both site and remote filesystems.

III. Throughput Profiles

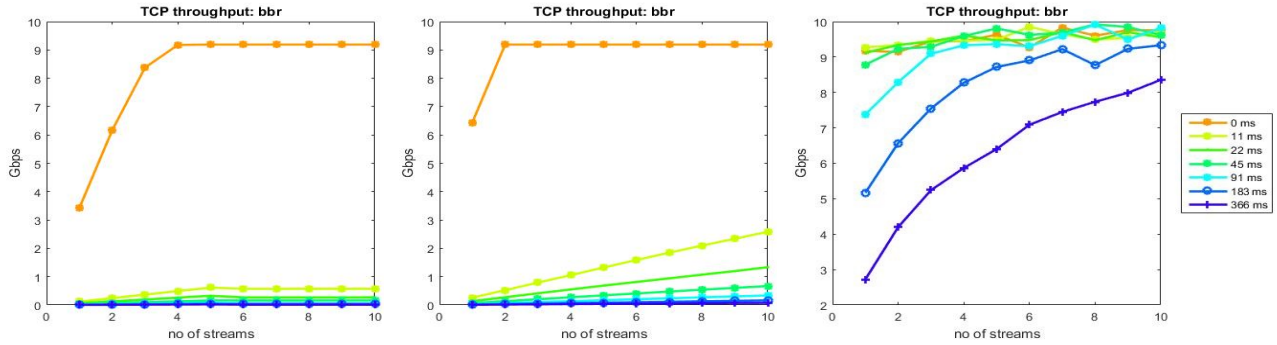
We have collected extensive UDT and TCP throughput measurements over connections of seven different lengths, using either UDT or TCP variants (BBR, CUBIC, HTCP, STCP), and for the latter, three buffer sizes and 1–10 parallel streams. The connection throughput profiles share certain common qualitative properties in terms of monotonicity and concave/convex regions, which we summarize in this section; more detailed descriptions and analyses of these large data sets are provided in [12] and [22].

A. UDT and TCP Throughput Measurement Profiles

We compute the mean throughput profile by taking the average throughput rates from repeated transfer experiments conducted at specific RTT (τ) values for each TCP variant and numbers of parallel streams. For each time trace, we sample throughput for a total duration of 100 seconds at one second intervals, and repeat this process 10 or 100 times for each RTT. We estimate the average throughput for each sampled trace that provides multiple throughput estimates at each RTT, which together constitute the aggregate throughput profile.

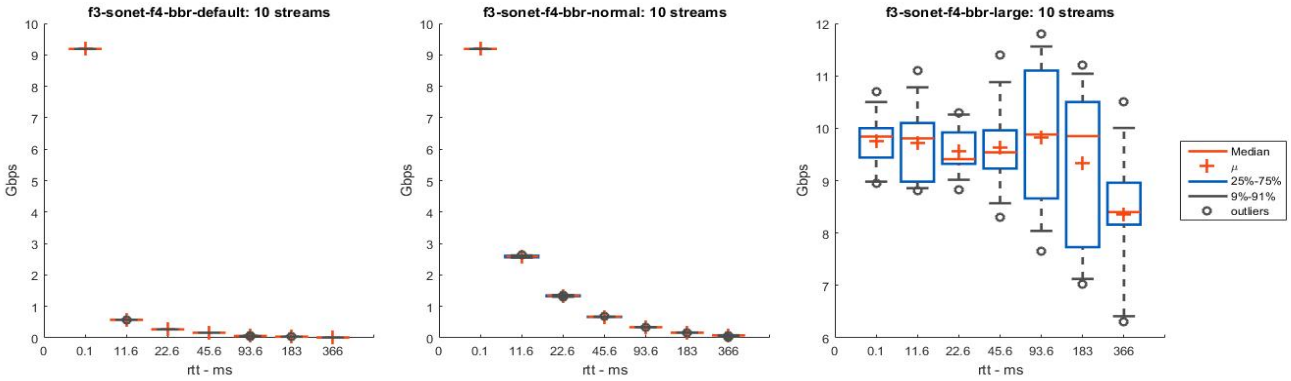
The aggregate UDT throughput profile in Fig. 3 shows the throughput mean (denoted as "+"), quartiles, minima and maxima. We examine both the standard packet size of 1,500 bytes (as specified by the current protocol) and for 9,000 byte jumbograms. Such 9,000 byte frames have been shown to reduce overheads and CPU cycles, and thereby improve the communication efficiency over 10 Gbps connections. The profile box plots indeed show that compared to the default packet size, jumbograms effectively lead to higher throughputs that are sustained for connections with RTTs above 100 ms.

In Fig. 4, mean throughput rates of BBR are plotted for three



(a) default (b) normal (c) large

Fig. 4: Throughput with variable RTTs, number of streams, and buffer sizes for BBR



(a) default (b) normal (c) large

Fig. 5: Throughput box plots with variable RTTs and buffer sizes for BBR with 10 streams

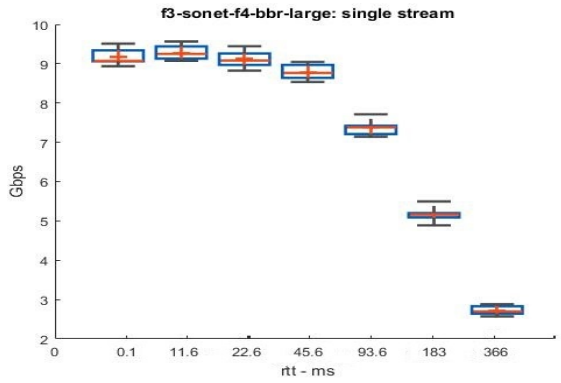


Fig. 6: Throughput box plots with variable RTTs for single-stream BBR with large buffers

buffer sizes, namely, default, normal, and large. A large buffer size significantly improves the mean throughput, especially for longer connections; for instance, the throughput of 10 streams for 366 ms RTT improves from under 100 Mbps to over 8 Gbps as the buffer size increases. Unless otherwise specified, subsequent discussions primarily address performance with large buffers. As seen from Fig. 5, for BBR with large buffers and 10 parallel streams, using the default and normal buffer sizes results in entirely convex profiles; with the large buffer size, a concave profile is observed for the mean throughput, as shown in Fig. 5(c) for the 10 stream case and in Fig. 6 for the single stream case. Comparing these BBR profiles with the corresponding profiles for CUBIC, HTCP, and STCP in [22], we observe that BBR generally requires much larger buffers; namely, large buffers are required for BBR to achieve similar throughput rates under "default" or "normal" buffer sizes in the other TCP variants. In addition, Fig. 5(c) shows the mean throughput values of BBR with 10 parallel streams and large buffers are higher than those achieved using other TCP variants, albeit with much larger variations.

B. Throughput Model

To explain the overall qualitative behavior observed in measurements as discussed in the previous subsection, a generic and coarse throughput model is proposed for TCP [22], [23] and UDP [12]. The throughput dynamics with fixed parameters over a connection with RTT τ and capacity C are characterized by two phases:

(a) *Ramp-Up Phase:* In the ramp-up phase, the throughput increases for a duration of T_R until it reaches a peak, and then switches to a sustained throughput phase. The average

throughput in this phase is

$$\bar{\theta}_R(\tau) = \frac{1}{T_R} \int_0^{T_R} \theta(\tau, t)\, dt\,.$$

(b) *Sustained Throughput Phase:* Once the throughput reaches the peak, it is "sustained" using a mechanism that processes the acknowledgments, and infers and responds to losses. The average throughput in this phase is

$$\bar{\theta}_S(\tau) = \frac{1}{T_S} \int_{T_R}^{T_R+T_S} \theta(\tau, t)\, dt\,.$$

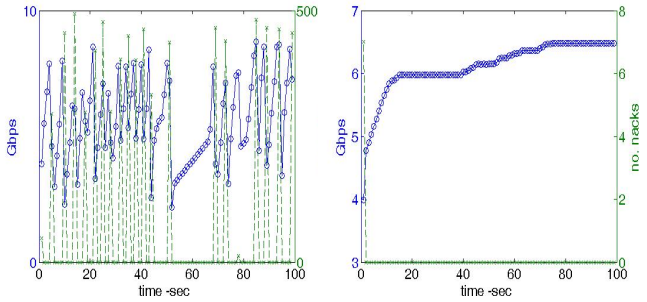
In general, the average $\bar{\theta}_S$ lies below the link capacity due to variations in throughput as shown in Fig. 1(b).

Using $\bar{\theta}_R$ and $\bar{\theta}_S$, the average throughput is given as

$$\Theta_O(\tau) = \frac{T_R}{T_O}\bar{\theta}_R(\tau) + \frac{T_S}{T_O}\bar{\theta}_S(\tau) = \bar{\theta}_S(\tau) - f_R\left(\bar{\theta}_S(\tau) - \bar{\theta}_R(\tau)\right),$$

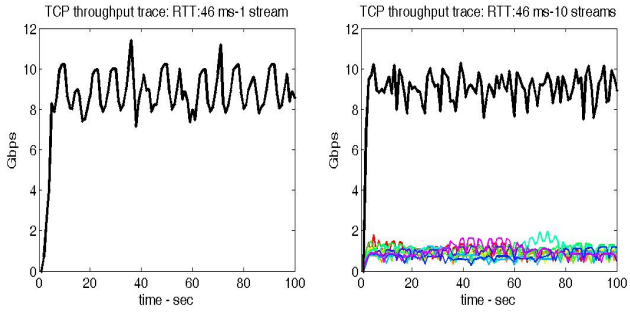
where $T_O = T_R + T_S$ and $f_R = T_R/T_O$. For a large observation period T_O with a fast ramp-up, typical in small RTT (τ) settings, the qualitative properties of $\theta_S(\tau)$ directly carry over to $\Theta_O(\tau)$. On the other hand, for large RTT, the ramp-up period can be significantly longer, for example, 10 seconds for 366 ms RTT, and therefore the difference term $\bar{\theta}_S - \bar{\theta}_R$ modulates the behavior of $\theta_S(\tau)$ in determining that of $\Theta_O(\tau)$. If $\theta_S(t) \approx C$, then $\Theta_O(\tau)$ decreases with τ since $\bar{\theta}_R \leq C$, and $\bar{\theta}_S - \bar{\theta}_R > 0$. However, if throughput falls much below C after the ramp-up and has significant random variations when repeated, it is quite possible for $\Theta_O(\tau)$ to increase with respect to τ in certain albeit small regions, for example, lower RTT regions in the UDT profile of Fig. 3.

Now, a concave throughput profile with respect to RTT is desirable since throughput is sustained as τ increases. The concave to convex transition point for the profiles is associated with an exponential increase in ramp-up followed by sustained throughput, $\theta_S(t) \approx C$; either a slower ramp-up or an unsustained peak throughput can lead to non-concave profiles. In most of the measured throughput profiles shown in the previous subsection, the profile is concave when RTT is small, and at a certain transition point it becomes and continues to be convex as RTT increases. This behavior is in part a result of various host buffers being sufficiently large to fill up the connection to the near capacity C combined with the fast response of TCP congestion control at lower RTT. Furthermore, it can also be shown qualitatively that adopting larger buffer sizes and/or more parallel streams leads to expanded concave regions and higher overall throughput [22]. In contrast, traditional TCP models, driven primarily by losses, result in throughput profiles having the generic form $\hat{\mathcal{T}}(\tau) = a + b/\tau^c$, for suitable constants a, b, and $c \geq 1$ [28]. These convex profiles (since $\frac{d\mathcal{T}}{d\tau} = -b/\tau^2$ increases with τ) are typical of large transfers and longer RTTs, and do not adequately account for transfers that lead to concave portions in the measured profiles.



(a) 22.6 ms RTT, trace 1 (b) 22.6 ms RTT, trace 2

Fig. 7: Sample send rate and NACK traces for UDT



(a) 1 stream (b) 10 streams

Fig. 8: Throughput traces for CUBIC with large buffers, 45.6 ms RTT, and variable numbers of streams

IV. Dynamics of Throughput Measurements

The throughput traces provide more detailed information about the transfer processes than the mean throughput profiles. As before, we sample the transfer rates at one-second intervals for a total duration of 100 seconds. However, in contrast to the experiments described in the previous section, the transfer size here is not fixed, and a higher average throughput indicates a larger transfer size.

Fig. 7 shows two sample time traces of UDT throughput measurement (along with the number of NACKs) for a 22.6 ms RTT connection. Large NACK counts, often in the hundreds, accompany each drop in the send rate; and such drops can occur frequently, as shown in Fig. 7(a). In fact, most traces at lower RTTs exhibit such oscillations over time. Nevertheless, there also exist traces, such as that shown in Fig. 7(b), where losses are few during a 100 second run. With increasing RTTs, such steady behavior becomes more common.

Fig. 8 show time traces for CUBIC throughput measurements from [22] over a 45.6 ms RTT connection, with 1 and 10 streams and large buffers. In these plots, the thick black curves describe the aggregate transfer rates, whereas different colored curves correspond to rates for individual streams. As can be seen from these plots, while the per-stream transfer rate decreases with more streams, the aggregate rates appear to hover around 9 Gbps and the transfer size is around 100 GB for most cases.

We can visualize the stability profile of these throughput traces via Poincaré maps. A *Poincaré map* $M : \Re_d \longmapsto \Re_d$

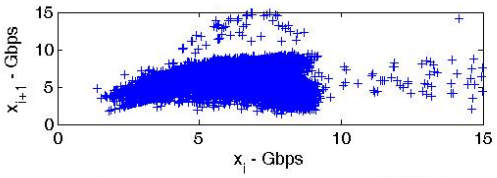
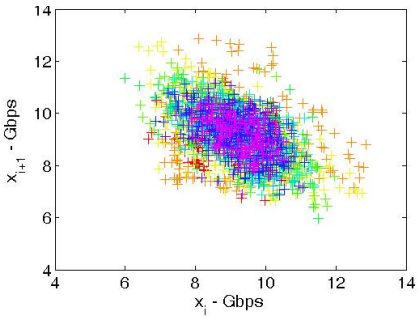
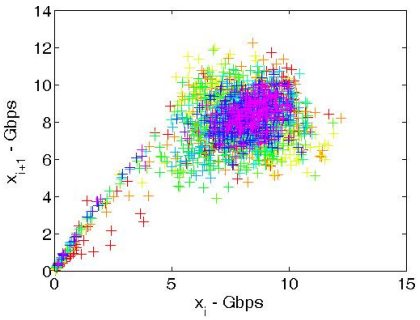


Fig. 9: Poincaré map for UDT with 22.6 ms RTT



(a) 11.6 ms



(b) 183 ms

Fig. 10: Aggregate Poincaré maps for CUBIC with large buffers and 11.6 ms or 183 ms RTTs

specifies a sequence, called the trajectory, of a real-vector state $X_i \in \Re_d$ that is updated at each time step i such that $X_{i+1} = M(X_i)$ [2]. The trajectories generated by a Poincaré map M are characterized by the *Lyapunov Exponent*, defined as $\mathcal{L}_M = \ln \left| \frac{dM}{dX} \right|$, which describes the separation of the trajectories that originate from the nearby states. In particular, negative Lyapunov exponents correspond to stable system dynamics [2].

The theoretical Poincaré map of the UDT analytical model is a simple monotonically increasing function that flattens out at peak throughput values [12]. Fig. 9 shows the Poincaré map for UDT estimated from traces for 22.6 ms RTT. It is evident that the sample points have covered most of the bottom left area here (relative to the (10,10) point), indicating large variations in time traces, as seen in Fig. 7(a). Nevertheless, the Poincaré map estimated using measurements gradually becomes thinner with increasing RTTs and draws closer to the theoretical shape described above.

In terms of the aggregate transfer rate, in Poincaré maps for CUBIC in Fig. 10, each color represents a different stream count, and the points are superimposed on top of one another with varying flow counts, forming a cluster that describe the sustainment stage. In particular, the 183 ms RTT case demonstrates the effect of a longer ramp-up stage by the points from the origin leading up to the cluster, which is absent in the 11.6 ms RTT case. Interestingly, the "tilts" of the two clusters also appear different: the 183 ms RTT cluster in Fig. 10(b) aligns more with the ideal 45° line, whereas the 11.6 ms RTT cluster in Fig. 10(a) tilts to the left, indicating a less stable profile of the corresponding time traces (even with overall higher mean throughput rates). This analysis is further confirmed by the Lyapunov exponents [22], whose values in the 183 ms RTT case form a more compact cluster closer to the zero line as opposed to the 11.6 ms RTT case. In addition, it was shown that more streams and larger buffers reduce the instability in aggregate transfer rates by pulling the Lyapunov exponents closer to zero (details can be found in [22]).

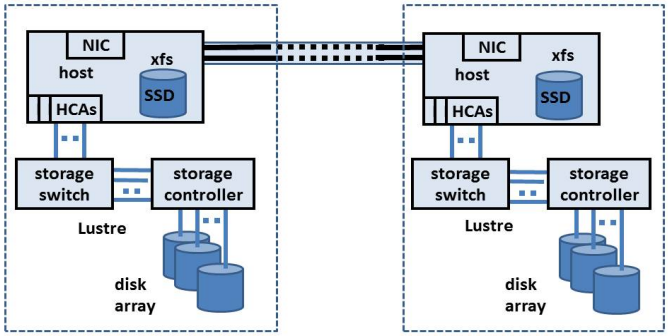


Fig. 11: File transfers over long-haul connections

V. File Transfer Measurements

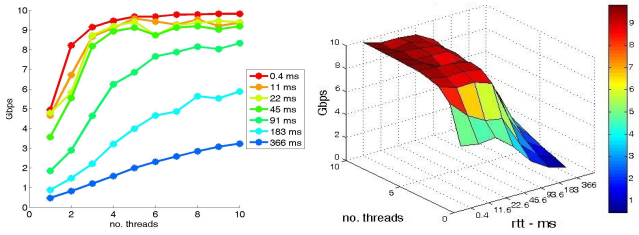
A wide-area disk-to-disk file transfer encompasses storage devices, data transfer hosts, and local- and wide-area connections as illustrated in Fig. 11. Major sites often use dedicated data transfer hosts, such as DTNs, with high performance Network Interface Cards (NICs) to access network connections and Host Channel Adapters (HCAs) to access network storage systems. Transfers also involve a range of software components, including filesystem I/O modules for disk access and the TCP/IP stack for network transport. When filesystems are mounted at individual sites, file transfer software such as GridFTP [29] and XDD [31] running on the hosts is used for transport, and detailed measurements and analyses of XDD transfers using our testbed are presented in [21]. Another method is to mount filesystems over wide-are networks [19], [30], wherein file transfers are handled by the underlying filesystem and a re transparent to the user.

A. XDD File Transfer Measurements

High-performance disk-to-disk transfers between filesystems at different sites require the composition of complex file I/O and network subsystems, and host orchestration. For example, as mentioned earlier, the Lustre filesystem employs multiple OSTs to manage collections of disks, multiple OSSes to stripe file contents, and distributed MDSes to provide site-wide file naming and access. However, sustaining high file-transfer rates requires *joint* optimization of subsystem

parameters to account for the impedance mismatches among them [25]. For Lustre filesystems, important parameters are the stripe size and number of stripes for the files, and these are typically specified at the creation time; the number of parallel I/O threads for read/write operations are specified at the transfer time. To sustain high throughput, I/O buffer size and the number of parallel threads are chosen to be sufficiently large, and as we will illustrate, this simple heuristic is not always optimal. For instance, wide-area file transfers over 10 Gbps connections between two Lustre filesystems achieve transfer rates of only 1.5 Gbps, when striped across 8 storage servers, accessed with 8 MB buffers, and with 8 I/O and TCP threads [21], even though peak network memory-transfer rate and local file throughput are each close to 10 Gbps.

We measured file I/O and network throughput and file-transfer rates over Lustre and XFS filesystems for a suite of seven emulated connections in the 0–366 ms RTT range, which are used for memory transfer measurements in Section III. We collected two sets of XDD disk-to-disk file transfer measurements, one from XFS to XFS and one from Lustre to Lustre, and each experiment was repeated 10 times. We considered both buffered I/O (the Linux default) and direct I/O options for Lustre. In the latter, XDD avoids the local copies of files on hosts by directly reading and writing into its buffers, which significantly improves the transfer rates. Results based on these measurements are summarized in [21]: (a) strategies of large buffers and higher parallelism do not always translate into higher transfer rates; (b) direct I/O methods that avoid file buffers at the hosts provide higher wide-area transfer rates, and (c) significant statistical variations in measurements, due to complex interactions of non-linear TCP dynamics with parallel file I/O streams, necessitate repeated measurements to ensure confidence in inferences based on them.

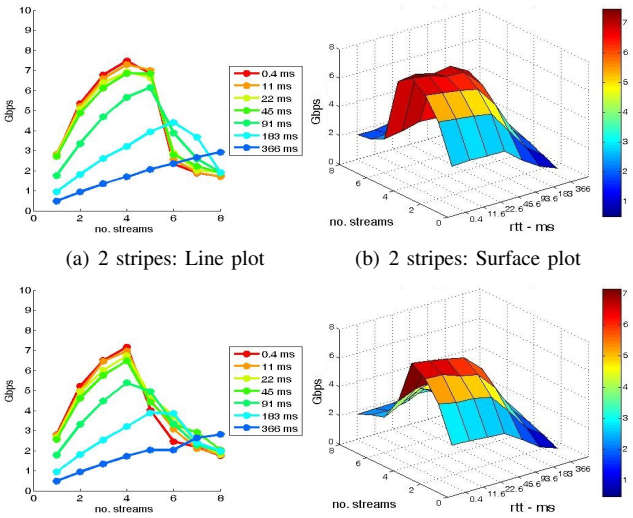


(a) Write rates: Line plot (b) Write rates: Surface plot

Fig. 12: Mean XFS file write rates

We first consider XFS-to-XFS transfers. From the aggregate mean throughput line plot in Fig. 12(a), we can see that throughput increases with the number of flows. For instance, whereas the mean throughput peaks at 5 Gbps with one flow, the peak (occurring with 0.4 ms RTT) rapidly jumps to above 9 Gbps with four flows, even closely approaching 10 Gbps with seven flows. Mean throughput generally decreases with RTT, consistent with most data transfer protocols. The surface plots in Fig. 12(b) indicate a *monotonically increasing* trend.

In the default I/O Lustre setup, the number of flows varies from 1 to 8, and the number of stripes is either 2 or 8. Fig. 13 shows the default I/O Lustre mean throughputs. Compared to XFS, the overall throughput here is much lower, especially for lower RTTs (such differences become less pronounced as RTT increases). For example, at lower RTTs, the mean throughput peaks at four flows, starts to decrease with five flows, and takes a nosedive at six flows. The sharp drop is delayed at higher RTTs, with throughput peaking at five flows for 91 ms RTT and at six flows for 183 ms RTT, and increasing all the way through eight flows for 366 ms RTT. The line plots Figs. 13(a) and 13(c) show that the sharp drop in throughput, if any, occurs earlier, at five flows, when eight stripes are used instead of two stripes. This result is also confirmed by the surface plots: the "dome" in Fig. 13(d) is narrower than that in Fig. 13(b).



(a) 2 stripes: Line plot (b) 2 stripes: Surface plot

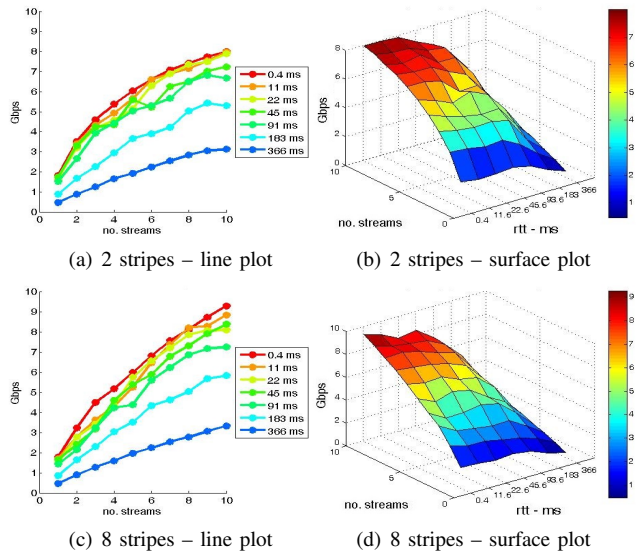
(c) 8 stripes: Line plot (d) 8 stripes: Surface plot

Fig. 13: Mean default I/O Lustre file write rates

We repeated the above experiments using direct I/O Lustre. The mean throughput plots in Fig. 14 show the overall monotonic trend as observed in the XFS results. In particular, single-flow throughput is lower that of default I/O Lustre, but the throughput steadily increases with the number of flows. Using two stripes yields somewhat higher transfer rates compared to eight stripes for lower flow counts. With more flows, overall throughput is higher, and eight stripes is the better option. Although the peak rates with 10 flows and eight stripes are lower compared to XFS, they are significantly higher than that with default I/O. In addition, The XFS surface in Fig. 12(b) is higher and increases faster at lower RTTs compared to those of the direct I/O Lustre in Figs. 14(b) and 14(d), as a result of rates starting higher with one flow and further improving and approaching the peak with additional flows.

In summary, file transfer rates for both XFS and Lustre are affected by the number of flows. The mean rate increases monotonically with number of flows for both XFS and direct I/O Lustre, but decreases beyond a certain point for default I/O Lustre. The number of stripes in Lustre seems to have less impact on transfer rate than does the number of flows. Additional measurements with request sizes of 65 MB and 145 MB

show that the default 8 MB selection used here consistently yields the best performance. This result is expected, as the smaller request size provides finer-resolution data chunks to TCP streams.



(a) 2 stripes – line plot (b) 2 stripes – surface plot

(c) 8 stripes – line plot (d) 8 stripes – surface plot

Fig. 14: Mean direct I/O Lustre file write rates

B. Lustre over Wide-Area Networks

There have been several works that extend Lustre filesystems over wide-area networks providing a number of desired features: (i) file transfers are natively supported by the copy operation, which obviates the need for file transfer tools, such as XDD, Aspera [3], and GridFTP [29]; and (ii) applications involving file operations can be supported transparently over wide-area networks. Typical site Lustre filesystems are mounted over IB networks, which are subject to 2.5 ms latency limitation of the IB flow control method; hardware-based IB range extenders can be used to overcome this limitation, although such a solution is not scalable due to high cost and the need to bookend each long-haul connection with a pair of these extenders. In this subsection, we present initial results that utilize LNet routers (implemented on Linux hosts) to extend Lustre over long-haul links with 0–366 ms RTT, without requiring any additional hardware. Previous results are limited to relatively shorter distances and somewhat limited range of latencies [9], [15].

We utilized Ethernet clients on remote servers to mount Lustre, and used LNet routers at Lustre site to route between site IB network and Ethernet LAN connected to WAN, as shown in Fig. 15. Under this scheme, no changes to the site Lustre system over IB network are needed, and hosts connected to both IB network and site Ethernet LAN are configured as LNnet routers. Initial proof-of-principle throughput measurements using IOzone writes are shown in Fig. 16 for 10GigE emulated connections between two pairs of hosts, namely, 48-core stand-alone systems used for data transfers, and 32-core hosts which are part of a compute cluster. The hosts are configured to use HTCP and their buffer sizes are

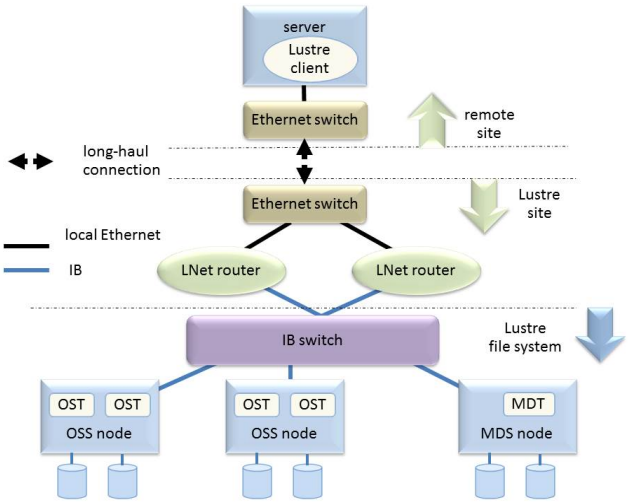
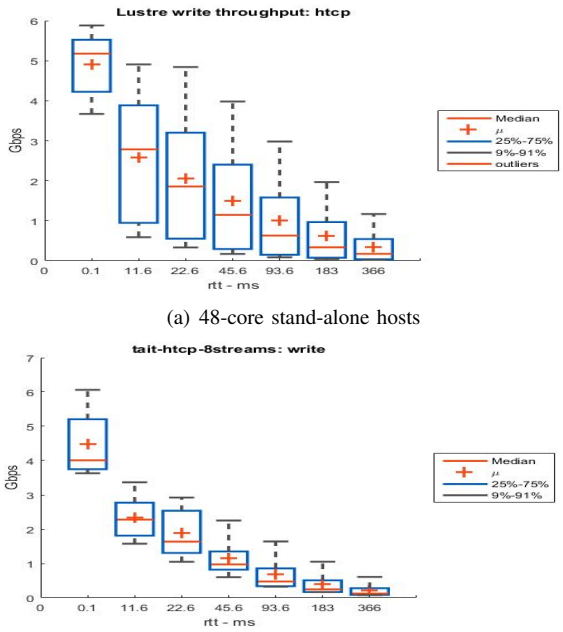


Fig. 15: Lustre over long-haul connections using LNet routers between local IB and Ethernet



(a) 48-core stand-alone hosts

(b) physical and emulated connections between hosts

Fig. 16: Throughput profiles of wide-area Lustre

set at largest allowable values. These throughput profiles are lower than the corresponding iperf profiles as expected, and the differences between the two types of hosts are within 10%. It is striking that the profiles are convex, indicative of buffer limits. Further investigations, including additional test configuration and examination of LNet routers configuration, are needed to improve the throughput performance.

VI. Conclusions and Open Areas

Wide-area data transfers in cloud computing and HPC scenarios are increasingly performed over dedicated network connections, driven in part by the promise of high throughput and stable transport dynamics. To study the performance of

UDT and TCP variants and their parameters for such transfers, we established a testbed and systematically collected measurements using physical and emulated dedicated connections. We have provided here a consolidated view of the results of these studies, some of which have been described separately in previous publications, and also reported on new results on BBR and Lustre filesystems mounted over wide-area networks. Our results reveal important concave-convex throughput profiles, and rich transport dynamics as indicated by the Poincaré map and Lyapunov exponents. These measurements and analyses enable the selection of an ideal high throughput transport method and parameters for a specific RTT range.

Further research advances are required before we can fully optimize data transfers over dedicated connections. We require (i) detailed analytical models that closely match the measurements of file and disk I/O throughput profiles, and (ii) enhanced first-principle UDT and TCP models to explain the dual-mode throughput profiles by integrating dynamics parameters, such as the Lyapunov exponents. Further measurements and analyses can provide more insights into the performance of BBR transport over dedicated connections, particularly when end systems are dissimilar in their computing, network, and I/O capacities. It is also of future interest to pursue learning-based approaches [13] to quantify the effect of various factors on throughput performance and develop SDN methods that use throughput profiles to select and set up suitable paths to match the transport protocols. Detailed experimentation with the LNet-based wide-area Lustre filesystem and investigation of its performance optimization is also of future interest. In general, the joint optimization of I/O, host, and network parameters to achieve peak and stable wide-area data transfers continues to be a challenge.

ACKNOWLEDGMENTS

This work is supported in part by the RAMSES and Net2013 projects, Office of Advanced Computing Research, U.S. Department of Energy, and by the Extreme Scale Systems Center, sponsored by U. S. Department of Defense, and performed at Oak Ridge National Laboratory managed by UT-Battelle, LLC for U.S. Department of Energy under Contract DE-AC05-00OR22725 and at Argonne National Laboratory under Contract DE-AC02-06CH11357.

REFERENCES

[1] Bay Microsystems. http://www.baymicrosystems.com.

[2] K. T. Alligood, T. D. Sauer, and J. A. Yorke. *Chaos: An Introduction to Dynamical Systems*. Springer-Verlag Pub., Reading, MA, 1996.

[3] Aspera high-speed file transfer. http://www-03.ibm.com/software/products/en/high-speed-file-transfer.

[4] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson. BBR: Congestion based congestion control. *ACM Qeueue*, 14(5), Dec. 2016.

[5] Science DMZ: Data Transfer Nodes. https://fasterdata.es.net/science-dmz/DTN.

[6] Energy Sciences Network. http://www.es.net.

[7] Y. Gu and R. L. Grossman. UDT: UDP-based data transfer for high-speed wide area networks. *Computer Networks*, 51(7), 2007.

[8] M. Hassan and R. Jain. *High Performance TCP/IP Networking: Concepts, Issues, and Solutions*. Prentice Hall, 2004.

[9] R. Henschel, S. Simms, D. Hancock, S. Michael, T. Johnson, N. Heald, T. William, et al. Demonstrating lustre over a 100Gbps wide area network of 3,500km. In *High Performance Computing, Networking, Storage and Analysis (SC), 2012 International Conference on*, pages 1–8, Nov. 2012.

[10] S. Jain, A. Kumar, S. Mandal, et al. B4: Experience with a globally-deployed software defined Wan. *SIGCOMM Comput. Commun. Rev.*, 43(4):3–14, Oct. 2013.

[11] T. Kelly. Scalable TCP: Improving performance in high speed wide area networks. *Computer Communication Review*, 33(2):83–91, 2003.

[12] Q. Liu, N. S. V. Rao, C. Q. Wu, D. Yun, R. Kettimuthu, and I. Foster. Measurement-based performance profiles and dynamics of udt over dedicated connections. In *International Conference on Network Protocols*. Singapore, Nov. 2016.

[13] Z. Liu, P. Balaprakash, R. Kettimuthu, and I. T. Foster. Explaining wide area data transfer performance. In *ACM Symposium on High-Performance Parallel and Distributed Computing (HPDC)*, Washington, DC, Jun. 2017.

[14] M. Mathis, J. Semke, J. Mahdavi, and T. Ott. The mascroscopic behavior of the TCP congestion avoidance algorithm. *Computer Communication Review*, 27(3), 1997.

[15] S. Michael, L. Zhen, R. Henschel, S. Simms, E. Barton, and M. Link. A study of Lustre networking over a 100 gigabit wide area network with 50 milliseconds of latency. In *Proceedings of the fifth international workshop on Data-Intensive Distributed Computing*, page 4352, 2012.

[16] Obsidian Research Corporation, http://www.obsidianresearch.com.

[17] On-demand Secure Circuits and Advance Reservation System. http://www.es.net/oscars.

[18] J. Padhye, V. Firoiu, D. F. Towsley, and J. F. Kurose. Modeling TCP Reno performance: A simple model and its empirical validation. *IEEE/ACM Transactions on Networking*, 8(2):133–145, 2000.

[19] J. Palencia, R. Budden, and K. Sullivan. Kerberized Lustre 2.0 over the WAN. In *Proceedings of the 2010 TeraGrid Conference*, Aug. 2010.

[20] N. S. V. Rao, J. Gao, and L. O. Chua. On dynamics of transport protocols in wide-area internet connections. In *Complex Dynamics in Communication Networks*. Springer-Verlag Publishers, 2005.

[21] N. S. V. Rao, Q. Liu, S. Sen, G. Hinkel, N. Imam, B. W. Settlemyer, I. T. Foster, et al. Experimental analysis of file transfer rates over wide-area dedicated connections. In *18th IEEE International Conference on High Performance Computing and Communications (HPCC)*, pages 198–205, Sydney, Australia, Dec. 2016.

[22] N. S. V. Rao, Q. Liu, S. Sen, D. Towsley, G. Vardoyan, I. T. Foster, and R. Kettimuthu. TCP throughput profiles using measurements over dedicated connections. In *ACM Symposium on High-Performance Parallel and Distributed Computing (HPDC)*, Washington, DC, Jun. 2017.

[23] N. S. V. Rao, D. Towsley, G. Vardoyan, B. W. Settlemyer, I. T. Foster, and R. Kettimuthu. Sustained wide-area TCP memory transfers over dedicated connections. In *IEEE International Conference on High Performance and Smart Computing*, New York, NY, Aug. 2015.

[24] I. Rhee and L. Xu. CUBIC: A new TCP-friendly high-speed TCP variant. In *Proceedings of the Third International Workshop on Protocols for Fast Long-Distance Networks*, 2005.

[25] B. W. Settlemyer, J. D. Dobson, S. W. Hodson, J. A. Kuehn, S. W. Poole, and T. M. Ruwart. A technique for moving large data sets over high-performance long distance networks. In *IEEE 27th Symposium on Mass Storage Systems and Technologies (MSST)*, pages 1–6, May 2011.

[26] T. Shanley. *InfiniBand Network Architecture*, volume I and II. MindShare, Inc., 2003.

[27] R. N. Shorten and D. J. Leith. H-TCP: TCP for high-speed and long-distance networks. In *Proceedings of the Third International Workshop on Protocols for Fast Long-Distance Networks*, 2004.

[28] Y. Srikant and L. Ying. *Communication Networks: An Optimization, Control, and Stochastic Networks Perspective*. Cambridge University Press, 2014.

[29] GT 4.0 GridFTP. http://www.globus.org.

[30] J. Walgenbach, S. C. Simms, K. Westneat, and J. P. Miller. Enabling Lustre WAN for production use on the TeraGrid: a lightweight UID mapping scheme. In *Proceedings of the 2010 TeraGrid Conference*, Aug. 2010.

[31] XDD - The eXtreme dd toolset, https://github.com/bws/xdd.

[32] T. Yee, D. Leith, and R. Shorten. Experimental evaluation of high-speed congestion control protocols. *Transactions on Networking*, 15(5):1109–1122, 2007.